

Amendments to the Claims:

This listing of claims will replace all prior versions, and listing of claims in the application:

Listing of the Claims:

1. **(Currently Amended)** A method of loading a class file into a virtual machine, ~~said class file being associated with an object-oriented class, and said virtual machine operating in an object-oriented computing system,~~ said method comprising:

determining whether one or more components of said class **file** have been marked **in said class file for loading** ~~to be loaded~~ into said virtual machine; and

loading said one or more components of said class **file** into said virtual machine when said determining determines that said one or more components of said class **file** have been marked **in said class file for loading** ~~to be loaded~~ into said virtual machine; **and**

not loading one or more other components of said class when said determining determines that said other one or more components of said class have not been marked to be loaded into said virtual machine.

2. (Canceled)

3. **(Currently Amended)** A method as recited in claim 1, wherein said method further comprises:

marking one or more components of said class for loading into said virtual machine.

4. **(Currently Amended)** A method as recited in claim 3,

wherein said marking is done by defining a[[n]] **load** attribute **in said class file that indicates that one or more components of said class file have been selected for loading into said virtual machine** ~~associated with said object-oriented class, and~~

wherein said determining operates to determine whether one or more components of said class have been marked to be loaded into said virtual machine based on said **load** attribute ~~associated with said object-oriented class.~~

5. (Canceled)

6. **(Currently Amended)** A method as recited in claim ~~[[5]]~~ **4**, wherein said **load** attribute portion in said class file is implemented as an attribute table.

7. **(Currently Amended)** A method as recited in claim 5, wherein said attribute table includes **one or more** offsets of one or more components of the **said** class file with respect to said class file, **wherein said one or more offsets can be used to determine the location of said one or more components in said class file, thereby allowing said one or more components to be loaded into said virtual machine based on said one or more offsets.**

8. (Original) A method as recited in claim 6, wherein said determining whether **of** one or more components of said class have been marked to be loaded into said virtual machine comprises:

initiating a first sequential read of said class file; and

determining whether an attribute table has been found for said class file.

9. **(Currently Amended)** A method as recited in claim 8, wherein said method further comprises:

initiating a second sequential read of said class file;

determining, **during said second sequential read,** whether a component **of said class file** has been encountered;

and determining whether of said class file a component has a corresponding entry in said attribute table of said class file when said component has been encountered.

10. (Original) A class file suitable for loading into a virtual machine, said class file being associated with an object-oriented class, and said virtual machine operating in an object oriented computing system, said class file comprising:

a load attribute portion that includes information about one or more components of said class which have been marked to be loaded into said virtual machine.

11. (Original) A class file as recited in claim 10, wherein said attribute portion includes an attribute table.

12. (Original) A class file as recited in claim 11, wherein said attribute table includes offsets of one or more components of said class file.

13. (Currently Amended) A computer readable media including computer readable code for representing a class file suitable for loading into a virtual machine, said class file being associated with an object-oriented class, and said virtual machine operating in an object-oriented computing system,

wherein said computer readable code representing said class file comprises computer readable code representing a load attribute portion of said class file, and

wherein said attribute portion represents information about one or more components of said class that have been marked to be loaded into said virtual machine.

14. (Original) A computer readable media as recited in claim 13, wherein said attribute portion represents a load attribute table.

15. (Original) A computer readable media as recited in claim 13, wherein said attribute table includes representation of offsets associated with one or more components of said class file.

16. (Original) A method of loading a class file into a virtual machine, said class file being associated with an object-oriented class, and said virtual machine operating in an object-oriented computing system, said method comprising:

providing a load attribute for said class file;

associating one or more components of said class file with said load attribute to indicate that said one or more components of said class file are to be loaded; and

loading only said one or more components of said class file into said virtual machine.

17. (Original) A method as recited in claim 16, wherein said providing of said load attribute operates to provide an attribute table in said class file.

18. (Original) A method as recited in claim 17, wherein said attribute table includes offsets of one or more components of the class file with respect to said class file.

19. (Original) A method as recited in claim 18, wherein said method further comprises:

determining whether at least one component of said class file has been associated with said load attribute.

20. (Original) A method as recited in claim 18, wherein said determining operates to search said attribute table for an offset associated with said at least one component of said class file.

21. (Original) A method as recited in claim 16, wherein said determining comprises:
- initiating a first sequential read of said class file to determine whether said class file has an attribute table;
 - reading said attribute table when said class file has an attribute table; initiating a second read of said class file; and
 - determining whether at least one component of said class file has been associated with said load attribution.